

CLAIMS

We claim:

- 5 1. A computer memory structure comprising:
 a configuration object, for a managed
product, including:
 a key field; and
 a setting object pointer attribute.
- 10 2. The computer memory structure of Claim 1
 wherein said key field comprises a name field.
- 15 3. The computer memory structure of Claim 1
 wherein said configuration object further comprises:
 a sequence name field.
- 20 4. The computer memory structure of Claim 2
 wherein said configuration object further comprises:
 a sequence name field.
- 25 5. The computer memory structure of Claim 4
 wherein a first value is stored in said name field and
 a second value is stored in said sequence name field
 and further wherein said first and second values are a
same value.
- 30 6. The computer memory structure of Claim 1
 wherein said configuration object further comprises:
 a sequence revision field.
- 35 7. The computer memory structure of Claim 6
 wherein a timestamp for said configuration object is
 stored in said sequence revision field.

8. The computer memory structure of Claim 1 further comprising:

5 a setting object wherein said setting object is addressed by a pointer of said setting object pointer attribute.

9. The computer memory structure of Claim 8 wherein said setting object further comprises:

10 a key field.

10. The computer memory structure of Claim 8 wherein said setting object further comprises:

15 a setting data field.

11. The computer memory structure of Claim 10 wherein said setting data field comprises a setting text field.

20 12. The computer memory structure of Claim 9 wherein said key field comprises a setting identifier field.

25 13. The computer memory structure of Claim 8 wherein said setting object further comprises:
a sequence name field.

30 14. The computer memory structure of Claim 12 wherein said setting object further comprises:
a sequence name field.

35 15. The computer memory structure of Claim 14 wherein a first value is stored in said sequence identifier field and a second value is stored in said sequence name field and further wherein said first and second values are a same value.

16. The computer memory structure of Claim 8
wherein said setting object further comprises:
a sequence revision field.

5

17. The computer memory structure of Claim 16
wherein a timestamp for said setting object is stored
in said sequence revision field.

10 18. The computer memory structure of Claim 1
wherein said configuration object further comprises:
a parent configuration object pointer
attribute wherein upon said parent configuration
object pointer attribute including a pointer to
15 another configuration object, said configuration
object is a child configuration object.

19. The computer memory structure of Claim 18
wherein said pointer to another configuration object
20 comprises a distinguished name pointer.

20. The computer memory structure of Claim 8
wherein said pointer stored in said setting object
pointer attribute is a distinguished name pointer.

25

21. A computer memory structure comprising:
a configuration object, for a software
feature of a managed product, comprising:
a name field, wherein said name field is
30 a key field for said configuration object;
a sequence name field; and
a sequence revision field.

22. The computer memory structure of Claim 21
35 wherein said configuration object further comprises:

a pointer attribute for a pointer to a
setting object.

23. The computer memory structure of Claim 21
5 wherein a first value is stored in said name field and
a second value is stored in said sequence name field
and further wherein said first and second values are a
same value.

10 24. The computer memory structure of Claim 21
wherein a first value is stored in said name field and
a second value is stored in said sequence name field
and further wherein said first value is said second
value combined with a value in said sequence revision
15 field.

25. The computer memory structure of Claim 21
wherein said configuration object further comprises:
a pointer attribute for a pointer to a parent
20 configuration object.

26. A computer memory structure comprising:
a setting object, for a setting for a
software feature of a managed product, comprising:
25 a setting identifier field, wherein said
setting identifier field is a key field for
said setting object;
a sequence name field;
a sequence revision field; and
30 a setting text field.

27. The computer memory structure of Claim 26
wherein a first value is stored in said sequence
identifier field and a second value is stored in said
35 sequence name field and further wherein said first and
second values are a same value.

28. The computer memory structure of Claim 26 wherein a timestamp for said setting object is stored in said sequence revision field.

5

29. A method comprising:

specifying a configuration for a managed product using a configuration object; and

10 representing a modification to said configuration for said managed product using a derived configuration object of said configuration object.

30. The method of Claim 29 wherein said
15 configuration object and said derived configuration object comprise a configuration object inheritance chain.

31. The method of Claim 30 further comprising:
20 processing said configuration object inheritance chain to obtain an effective configuration for said managed product.

32. The method of Claim 31 wherein said
25 processing said configuration object inheritance chain comprises:

using a parent-child inheritance merge process.

30 33. A method comprising:
using a string in a setting object to specify a setting for a managed product; and
linking said setting object to a first configuration object for said managed product.

35

34. The method of Claim 33 further comprising:

generating a second configuration object for
said managed product.

35. The method of Claim 34 wherein said first
5 configuration object comprises:
a first memory structure comprising:
a first name field storing a name wherein
said name is a key for said first configuration
object; and
10 a first sequence revision field storing a
first timestamp for said configuration object.

36. The method of Claim 35 wherein said
generating a second configuration object further
15 comprising:
creating a second memory structure having a
second name field and a second sequence revision
field.

20 37. The method of Claim 36 wherein said
generating a second configuration object further
comprises:
copying said first timestamp from said first
sequence revision field to said second sequence
25 revision field.

38. The method of Claim 36 wherein said
generating a second configuration object further
comprises:
30 storing a second name in said second name
field wherein said second name field comprises a
combination of said name and said first time
stamp.

35 39. The method of Claim 35 further comprises:

overwriting said first timestamp in said first sequence revision field with a second timestamp.

5 40. The method of Claim 38 further comprises:
 overwriting said first timestamp in said first sequence revision field with a second timestamp.

10 41. The method of Claim 33 wherein using said string comprises using an extensible markup language string.

 42. The method of Claim 41 wherein said using
15 said string further comprises:
 using a name attribute with a namespecifier in a start tag in said string.

 43. The method of Claim 42 where said
20 namespecifier appends a literal name to a name of said start tag.

 44. The method of Claim 42 where said
 namespecifier appends current element text to a name of
25 said start tag.

 45. The method of Claim 42 where said
 namespecifier appends a current element attribute value
 to a name of said start tag.

30 46. The method of Claim 42 where said
 namespecifier appends a name of a subelement tag to a name of said start tag.

47. The method of Claim 42 where said
namespecifier appends text of a subelement to a name of
said start tag.

5 48. The method of Claim 42 where said
namespecifier appends a subelement attribute value to a
name of said start tag.

49. A method comprising;
10 generating an effective configuration for a
managed product from a configuration inheritance
chain.

50. The method of Claim 49 wherein said
15 generating an effective configuration comprises:
getting a mark-up language string for a most-
derived configuration object.

51. The method of Claim 50 wherein said
20 generating an effective configuration further
comprises:
converting said mark-up language string for
said most-derived configuration object to a
derived tree structure having nodes wherein a
25 plurality of nodes in said derived tree structure
include collision detection names.

52. The method of Claim 51 wherein a collision
detection name for a node in said plurality of nodes is
30 a name of a start tag when said start tag does not
include a name attribute.

53. The method of Claim 51 wherein a collision
detection name for a node in said plurality of nodes is
35 combination of a name of a start tag and a string

determined by a namespecifier when said start tag includes a name attribute with said namespecifier.

54. The method of Claim 51 wherein said
5 generating an effective configuration comprises:
getting a mark-up language string for a
parent configuration object of said most-derived
configuration object.

10 55. The method of Claim 54 wherein said
generating an effective configuration further
comprises:
converting said mark-up language string for
said parent configuration object to a base tree
15 structure having nodes wherein a plurality of
nodes in said base tree structure include
collision detection names.

56. The method of Claim 55 wherein said
20 generating an effective configuration further
comprises:
combining said derived tree structure and
said base tree structure, by resolving at least
one collision between a node in the derived tree
25 structure having a collision detection name and a
node in the base tree structure having said
collision detection name, to form a merged tree
structure.

30 57. The method of Claim 56 wherein resolving at
least one collision between a node in the derived tree
structure having a collision detection name and a node
in the base tree structure having said collision
detection name further comprises;
35 merging said nodes to form a node of said
merged tree when said nodes have child nodes.

58. The method of Claim 56 wherein resolving at least one collision between a node in the derived tree structure having a collision detection name and a node
5 in the base tree structure having said collision detection name further comprises:
 copying said node in the derived tree structure to said merged tree when said nodes are leaf nodes.
- 10
59. The method of Claim 56 wherein resolving at least one collision between a node in the derived tree structure having a collision detection name and a node in the base tree structure having said collision
15 detection name further comprises;
 selecting a combination of said nodes to form a node of said merged tree based upon a value of a collision resolution mode attribute in a start tag for an element corresponding to one of said nodes.
- 20
60. The method Claim 59 where said value of said collision resolution mode attribute is merge.
61. The method Claim 59 where said value of said
25 collision resolution mode attribute is use base.
62. The method Claim 59 where said value of said collision resolution mode attribute is use derived.
- 30
63. The method Claim 59 where said value of said collision resolution mode attribute is accumulate.
64. The method of Claim 50 wherein said getting a mark-up language string for a most-derived
35 configuration object includes:

collapsing sibling elements with identical values of a name attribute into a single element.

- 5 65. A method comprising
 using an extensible markup language string in
a setting object to specify a setting for a
managed product; and
 including a name attribute in at least one
start tag in a XML string.
- 10 66. The method of Claim 65 further comprising:
 including a collision resolution mode
attribute for at least one start tag in said XML
string.
- 15 67. The method Claim 66 where a value of said
collision resolution mode attribute is merge.
- 20 68. The method Claim 66 where a value of said
collision resolution mode attribute is use base.
69. The method Claim 66 where a value of said
collision resolution mode attribute is use derived.
- 25 70. The method Claim 66 where a value of said
collision resolution mode attribute is accumulate.
71. The method of Claim 65 wherein said name
attribute includes a namespecifier.
- 30 72. The method of Claim 71 where said
namespecifier appends a literal name to a name of said
start tag.

73. The method of Claim 71 where said
namespecifier appends current element text to a name of
said start tag.

5 74. The method of Claim 71 where said
namespecifier appends a current element attribute value
to a name of said start tag.

75. The method of Claim 71 where said
10 namespecifier appends a name of a subelement tag to a
name of said start tag.

76. The method of Claim 71 where said
namespecifier appends text of a subelement to a name of
15 said start tag.

77. The method of Claim 71 where said
namespecifier appends a subelement attribute value to a
name of said start tag.

20

78. A computer-program product comprising a
computer-readable medium containing computer program
code for a method comprising:

25 specifying a configuration for a managed
product using a configuration object; and
 representing a modification to said
configuration for said managed product using a
derived configuration object of said configuration
object.

30

79. A structure comprising:

 means for specifying a configuration for a
managed product using a configuration object; and
 means for representing a modification to said
35 configuration for said managed product using a

derived configuration object of said configuration object.

80. A computer-program product comprising a
5 computer-readable medium containing computer program
code for a method comprising:
using a string in a setting object to specify
a setting for a managed product; and
10 linking said setting object to a first
configuration object for said managed product.

81. A structure comprising:
means for using a string in a setting object
15 to specify a setting for a managed product; and
means for linking said setting object to a
first configuration object for said managed
product.

20 82. A structure comprising:
means for getting a mark-up language string
for a most-derived configuration object; and
means for converting said mark-up language
string for said most-derived configuration object
25 to a derived tree structure having nodes wherein a
plurality of nodes in said derived tree structure
include collision detection names.

83. The structure of Claim 82 further comprising:
30 means for getting a mark-up language string
for a parent configuration object of said most-
derived configuration object.

84. The structure of Claim 83 further comprising:
35 means for converting said mark-up language
string for said parent configuration object to a

base tree structure having nodes wherein a plurality of nodes in said base tree structure include collision detection names.

5 85. The structure of Claim 84 further comprising:
 means for combining said derived tree
 structure and said base tree structure, by
 resolving at least one collision between a node in
10 the derived tree structure having a collision
 detection name and a node in the base tree
 structure having said collision detection name, to
 form a merged tree structure.

 86. A computer-program product comprising a
15 computer-readable medium containing computer program
 code for a method comprising:
 using an extensible markup language string in
 a setting object to specify a setting for a
 managed product; and
20 including a name attribute in at least one
 start tag in said XML string.